



FILE ID SUBST

D 13

SSSSSSSSS UU UU BBBBBBBBBB SSSSSSSSS TTTTTTTTTT  
SSSSSSSSS UU UU BBBBBBBBBB SSSSSSSSS TTTTTTTTTT  
SS UU UU B B BB SS TT  
SSSSSS UU UU BBBBBBBBBB SSSSSSSSS TT  
SSSSSS UU UU BBBBBBBBBB SSSSSSSSS TT  
SS UU UU B B BB SS TT  
SSSSSSSS UUUUUUUUUUU BBBBBBBBBB SSSSSSSSS TT  
SSSSSSSS UUUUUUUUUUU BBBBBBBBBB SSSSSSSSS TT

The diagram illustrates a triangular arrangement of binary digits (LL, II, SS) in three columns. The left column contains 11 'LL' pairs, the middle column contains 11 'II' pairs, and the right column contains 11 'SS' pairs. Each row in the triangle represents a power of 2, starting from 1 at the top and increasing to 1024 at the bottom.

Row	Column 1 (LL)	Column 2 (II)	Column 3 (SS)
1	LL	II	SS
2	LL	II	SS
3	LL	II	SS
4	LL	II	SS
5	LL	II	SS
6	LL	II	SS
7	LL	II	SS
8	LL	II	SS
9	LL	II	SS
10	LL	II	SS
11	LL	II	SS
12	LL	II	SS
13	LL	II	SS
14	LL	II	SS
15	LL	II	SS
16	LL	II	SS
17	LL	II	SS
18	LL	II	SS
19	LL	II	SS
20	LL	II	SS
21	LL	II	SS
22	LL	II	SS
23	LL	II	SS
24	LL	II	SS
25	LL	II	SS
26	LL	II	SS
27	LL	II	SS
28	LL	II	SS
29	LL	II	SS
30	LL	II	SS
31	LL	II	SS
32	LL	II	SS
33	LL	II	SS
34	LL	II	SS
35	LL	II	SS
36	LL	II	SS
37	LL	II	SS
38	LL	II	SS
39	LL	II	SS
40	LL	II	SS
41	LL	II	SS
42	LL	II	SS
43	LL	II	SS
44	LL	II	SS
45	LL	II	SS
46	LL	II	SS
47	LL	II	SS
48	LL	II	SS
49	LL	II	SS
50	LL	II	SS
51	LL	II	SS
52	LL	II	SS
53	LL	II	SS
54	LL	II	SS
55	LL	II	SS
56	LL	II	SS
57	LL	II	SS
58	LL	II	SS
59	LL	II	SS
60	LL	II	SS
61	LL	II	SS
62	LL	II	SS
63	LL	II	SS
64	LL	II	SS
65	LL	II	SS
66	LL	II	SS
67	LL	II	SS
68	LL	II	SS
69	LL	II	SS
70	LL	II	SS
71	LL	II	SS
72	LL	II	SS
73	LL	II	SS
74	LL	II	SS
75	LL	II	SS
76	LL	II	SS
77	LL	II	SS
78	LL	II	SS
79	LL	II	SS
80	LL	II	SS
81	LL	II	SS
82	LL	II	SS
83	LL	II	SS
84	LL	II	SS
85	LL	II	SS
86	LL	II	SS
87	LL	II	SS
88	LL	II	SS
89	LL	II	SS
90	LL	II	SS
91	LL	II	SS
92	LL	II	SS
93	LL	II	SS
94	LL	II	SS
95	LL	II	SS
96	LL	II	SS
97	LL	II	SS
98	LL	II	SS
99	LL	II	SS
100	LL	II	SS
101	LL	II	SS
102	LL	II	SS
103	LL	II	SS
104	LL	II	SS
105	LL	II	SS
106	LL	II	SS
107	LL	II	SS
108	LL	II	SS
109	LL	II	SS
110	LL	II	SS
111	LL	II	SS
112	LL	II	SS
113	LL	II	SS
114	LL	II	SS
115	LL	II	SS
116	LL	II	SS
117	LL	II	SS
118	LL	II	SS
119	LL	II	SS
120	LL	II	SS
121	LL	II	SS
122	LL	II	SS
123	LL	II	SS
124	LL	II	SS
125	LL	II	SS
126	LL	II	SS
127	LL	II	SS
128	LL	II	SS
129	LL	II	SS
130	LL	II	SS
131	LL	II	SS
132	LL	II	SS
133	LL	II	SS
134	LL	II	SS
135	LL	II	SS
136	LL	II	SS
137	LL	II	SS
138	LL	II	SS
139	LL	II	SS
140	LL	II	SS
141	LL	II	SS
142	LL	II	SS
143	LL	II	SS
144	LL	II	SS
145	LL	II	SS
146	LL	II	SS
147	LL	II	SS
148	LL	II	SS
149	LL	II	SS
150	LL	II	SS
151	LL	II	SS
152	LL	II	SS
153	LL	II	SS
154	LL	II	SS
155	LL	II	SS
156	LL	II	SS
157	LL	II	SS
158	LL	II	SS
159	LL	II	SS
160	LL	II	SS
161	LL	II	SS
162	LL	II	SS
163	LL	II	SS
164	LL	II	SS
165	LL	II	SS
166	LL	II	SS
167	LL	II	SS
168	LL	II	SS
169	LL	II	SS
170	LL	II	SS
171	LL	II	SS
172	LL	II	SS
173	LL	II	SS
174	LL	II	SS
175	LL	II	SS
176	LL	II	SS
177	LL	II	SS
178	LL	II	SS
179	LL	II	SS
180	LL	II	SS
181	LL	II	SS
182	LL	II	SS
183	LL	II	SS
184	LL	II	SS
185	LL	II	SS
186	LL	II	SS
187	LL	II	SS
188	LL	II	SS
189	LL	II	SS
190	LL	II	SS
191	LL	II	SS
192	LL	II	SS
193	LL	II	SS
194	LL	II	SS
195	LL	II	SS
196	LL	II	SS
197	LL	II	SS
198	LL	II	SS
199	LL	II	SS
200	LL	II	SS
201	LL	II	SS
202	LL	II	SS
203	LL	II	SS
204	LL	II	SS
205	LL	II	SS
206	LL	II	SS
207	LL	II	SS
208	LL	II	SS
209	LL	II	SS
210	LL	II	SS
211	LL	II	SS
212	LL	II	SS
213	LL	II	SS
214	LL	II	SS
215	LL	II	SS
216	LL	II	SS
217	LL	II	SS
218	LL	II	SS
219	LL	II	SS
220	LL	II	SS
221	LL	II	SS
222	LL	II	SS
223	LL	II	SS
224	LL	II	SS
225	LL	II	SS
226	LL	II	SS
227	LL	II	SS
228	LL	II	SS
229	LL	II	SS
230	LL	II	SS
231	LL	II	SS
232	LL	II	SS
233	LL	II	SS
234	LL	II	SS
235	LL	II	SS
236	LL	II	SS
237	LL	II	SS
238	LL	II	SS
239	LL	II	SS
240	LL	II	SS
241	LL	II	SS
242	LL	II	SS
243	LL	II	SS
244	LL	II	SS
245	LL	II	SS
246	LL	II	SS
247	LL	II	SS
248	LL	II	SS
249	LL	II	SS
250	LL	II	SS
251	LL	II	SS
252	LL	II	SS
253	LL	II	SS
254	LL	II	SS
255	LL	II	SS
256	LL	II	SS
257	LL	II	SS
258	LL	II	SS
259	LL	II	SS
260	LL	II	SS
261	LL	II	SS
262	LL	II	SS
263	LL	II	SS
264	LL	II	SS
265	LL	II	SS
266	LL	II	SS
267	LL	II	SS
268	LL	II	SS
269	LL	II	SS
270	LL	II	SS
271	LL	II	SS
272	LL	II	SS
273	LL	II	SS
274	LL	II	SS
275	LL	II	SS
276	LL	II	SS
277	LL	II	SS
278	LL	II	SS
279	LL	II	SS
280	LL	II	SS
281	LL	II	SS
282	LL	II	SS
283	LL	II	SS
284	LL	II	SS
285	LL	II	SS
286	LL	II	SS
287	LL	II	SS
288	LL	II	SS
289	LL	II	SS
290	LL	II	SS
291	LL	II	SS
292	LL	II	SS
293	LL	II	SS
294	LL	II	SS
295	LL	II	SS
296	LL	II	SS
297	LL	II	SS
298	LL	II	SS
299	LL	II	SS
300	LL	II	SS
301	LL	II	SS
302	LL	II	SS
303	LL	II	SS
304	LL	II	SS
305	LL	II	SS
306	LL	II	SS
307	LL	II	SS
308	LL	II	SS
309	LL	II	SS
310	LL	II	SS
311	LL	II	SS
312	LL	II	SS
313	LL	II	SS
314	LL	II	SS
315	LL	II	SS
316	LL	II	SS
317	LL	II	SS
318	LL	II	SS
319	LL	II	SS
320	LL	II	SS
321	LL	II	SS
322	LL	II	SS
323	LL	II	SS
324	LL	II	SS
325	LL	II	SS
326	LL	II	SS
327	LL	II	SS
328	LL	II	SS
329	LL	II	SS
330	LL	II	SS
331	LL	II	SS
332	LL	II	SS
333	LL	II	SS
334	LL	II	SS
335	LL	II	SS
336	LL	II	SS
337	LL	II	SS
338	LL	II	SS
339	LL	II	SS
340	LL	II	SS
341	LL	II	SS
342	LL	II	SS
343	LL	II	SS
344	LL	II	SS
345	LL	II	SS
346	LL	II	SS
347	LL	II	SS
348	LL	II	SS
349	LL	II	SS
350	LL	II	SS
351	LL	II	SS
352	LL	II	SS
353	LL	II	SS
354	LL	II	SS
355	LL	II	SS
356	LL	II	SS
357	LL	II	SS
358	LL	II	SS
359	LL	II	SS
360	LL	II	SS
361	LL	II	SS
362	LL	II	SS
363	LL	II	SS
364	LL	II	SS
365	LL	II	SS
366	LL	II	SS
367	LL	II	SS
368	LL	II	SS
369	LL	II	SS
370	LL	II	SS
371	LL	II	SS
372	LL	II	SS
373	LL	II	SS
374	LL	II	SS
375	LL	II	SS
376	LL	II	SS
377	LL	II	SS
378	LL	II	SS
379	LL	II	SS
380	LL	II	SS
381	LL	II	SS
382	LL	II	SS
383	LL	II	SS
384	LL	II	SS
385	LL	II	SS
386	LL	II	SS
387	LL	II	SS
388	LL	II	SS
389	LL	II	SS
390	LL	II	SS
391	LL	II	SS
392	LL	II	SS
393	LL	II	SS
394	LL	II	SS
395	LL	II	SS
396	LL	II	SS
397	LL	II	SS
398	LL	II	SS
399	LL	II	SS
400	LL	II	SS
401	LL	II	SS
402	LL	II	SS
403	LL	II	SS
404	LL	II	SS
405	LL	II	SS
406	LL	II	SS
407	LL	II	SS
408	LL	II	SS
409	LL	II	SS
410	LL	II	SS

SUB  
V04  
SERIAL

```
1 0001 0 XTITLE 'Process <SUBSTITUTE flag>'  
2 0002 0 MODULE SUBST (  
3 0003 0 IDENT = 'V04-000'  
P 0004 0 XBLISS32[  
P 0005 0 ADDRESSING_MODE(INTERNAL=LONG_RELATIVE,NONEXTERNAL=LONG_RELATIVE)  
6 0006 0 ]  
7 0007 0 ) =  
8 0008 1 BEGIN  
9 0009 1  
10 0010 1 *****  
11 0011 1 *  
12 0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
13 0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
14 0014 1 * ALL RIGHTS RESERVED.  
15 0015 1 *  
16 0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
17 0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
18 0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
19 0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
20 0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
21 0021 1 * TRANSFERRED.  
22 0022 1 *  
23 0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
24 0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
25 0025 1 * CORPORATION.  
26 0026 1 *  
27 0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
28 0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
29 0029 1 *  
30 0030 1 *  
31 0031 1 *****  
32 0032 1 ++  
33 0033 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS  
34 0034 1 ABSTRACT: Processes the <SUBSTITUTE flag>  
35 0035 1  
36 0036 1  
37 0037 1  
38 0038 1  
39 0039 1 ENVIRONMENT: Transportable  
40 0040 1  
41 0041 1 AUTHOR: R.W.Friday CREATION DATE: May, 1978  
42 0042 1
```

SUBST  
V04-000

Process <SUBSTITUTE flag>  
Revision History

F 13  
16-Sep-1984 01:52:20    VAX-11 Bliss-32 v4.0-742  
14-Sep-1984 13:08:17    DISK\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1    Page 2

: 44  
45 0043 1 %SBTTL 'Revision History'  
46 0044 1  
47 0045 1 MODIFIED BY:  
48 0046 1  
49 0047 1 004 REM00004 Ray Marshall 30-Mar-1984  
50 0048 1 Added conditionals for foreign language support. Provided  
51 0049 1 German translations for the names of the months.  
52 0050 1  
53 0051 1 003 REM00003 Ray Marshall 07-Mar-1983  
54 0052 1 Global edit of all modules. Updated module names, idents,  
55 0053 1 copyright dates. Changed require files to BLISS library.  
56 0054 1  
0055 1 !--

```
58 0056 1 %SBTTL 'Module Level Declarations'  
59 0057 1  
60 0058 1 TABLE OF CONTENTS:  
61 0059 1  
62 0060 1  
63 0061 1 INCLUDE FILES:  
64 0062 1  
65 0063 1  
66 0064 1 LIBRARY 'NXPORT:XPORT';  
67 0065 1 REQUIRE 'REQ:RNODEF';  
68 0196 1  
69 U 0197 1 XIF DSRPLUS XTHEN  
70 U 0198 1 LIBRARY 'REQ:DPLLIB';  
71 0199 1 ! DSRPLUS BLISS Library  
72 0200 1 ELSE  
73 0201 1 LIBRARY 'REQ:DSRLIB';  
74 0201 1 ! DSR BLISS Library  
75 0201 1 XFI  
76 0202 1  
77 0203 1  
78 0204 1 MACROS:  
79 M 0205 1  
80 0206 1 MACRO  
81 0207 1 NAME_ENTRY(X) =  
82 0208 1    %HSPTR( (UPLIT ( %STRING( %CHAR(%CHARCOUNT(X)), X) ) ) );  
83 0209 1  
84 0210 1 EQUATED SYMBOLS:  
85 0211 1  
86 0212 1 LITERAL  
87 0213 1 FIRST_INDEX = 1,  
88 0214 1 SUBST_DATE = 1,  
89 0215 1 SUBST_TIME = 2,  
90 0216 1 SUBST_YEAR = 3,  
91 0217 1 SUBST_MONTH = 4,  
92 0218 1 SUBST_DAY = 5,  
93 0219 1 SUBST_HOURS = 6,  
94 0220 1 SUBST_MINUTES = 7,  
95 0221 1 SUBST_SECONDS = 8,  
96 0222 1 LAST_INDEX = 8;  
97 0223 1 !Keep updated for additional symbols!!!!  
98 0224 1 OWN STORAGE:  
99 0225 1  
100 0226 1 BIND  
101 0227 1 SUBST_NAMES = PLIT (NAME_ENTRY('DATE'),  
102 0228 1 NAME_ENTRY('TIME'),  
103 0229 1 NAME_ENTRY('YEAR'),  
104 0230 1 NAME_ENTRY('MONTH'),  
105 0231 1 NAME_ENTRY('DAY'),  
106 0232 1 NAME_ENTRY('HOURS'),  
107 0233 1 NAME_ENTRY('MINUTES'),  
108 0234 1 NAME_ENTRY('SECONDS')) : VECTOR;  
109 0235 1  
110 0236 1  
111 0237 1 EXTERNAL REFERENCES:  
112 0238 1  
113 0239 1  
114 0240 1 EXTERNAL  
115 0241 1 DATTIM : VECTOR,  
116 0242 1 IRA : FIXED_STRING,  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242
```

SUBST  
V04-000

Process <SUBSTITUTE flag>  
Module Level Declarations

H 13  
16-Sep-1984 01:52:20    VAX-11 Bliss-32 v4.0-742  
14-Sep-1984 13:08:17    DISK\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1 Page 4  
(3)

TIT  
V04

```
; 115 0243 1 KHAR;  
; 116 0244 1  
; 117 0245 1 EXTERNAL LITERAL                         !Error messages  
; 118 0246 1 RNFFNA,  
; 119 0247 1 RNFUDS;  
; 120 0248 1  
; 121 0249 1 EXTERNAL ROUTINE  
; 122 0250 1 CNVDAT,  
; 123 0251 1 CNVTIM,  
; 124 0252 1 ENDCHR,  
; 125 0253 1 ERMS.  
; 126 0254 1 GSLU  
; 127 0255 1 PACBÁS;
```

```
: 129
: 130
: 131
: 132
: 133
: 134
: 135
: 136
: 137
: 138
: 139
: 140
: 141
: 142
: 143
: 144
: 145
: 146
: 147
: 148
: 149
: 150
: 151
: 152
: 153
: 154
: 155
: 156
: 157
: 158
: 159
: 160
: 161
: 162
: 163
: 164
: 165
: 166
: 167
: 168
: 169
: 170
: 171
: 172
: 173
: 174
: 175
: 176
: 177
: 178
: 179
: 180
: 181
: 182
: 183
: 184
: 185

0256 1 %SBTTL 'SUBST --'
0257 1 GLOBAL ROUTINE subst : NOVALUE =
0258 1
0259 1 !++
0260 1 FUNCTIONAL DESCRIPTION:
0261 1
0262 1 See ABSTRACT, above.
0263 1
0264 1 FORMAL PARAMETERS: None
0265 1
0266 1 IMPLICIT INPUTS:
0267 1
0268 1 KHAR must be the current <SUBSTITUTE flag>, the first of
0269 1 the (presumed) pair.
0270 1
0271 1 IMPLICIT OUTPUTS: None
0272 1
0273 1 ROUTINE VALUE:
0274 1 COMPLETION CODES: None
0275 1
0276 1 SIDE EFFECTS: None
0277 1
0278 1 --
0279 1
0280 2 BEGIN
0281 2
0282 2 LOCAL
0283 2 DUMMY_FS : FIXED_STRING[50],
0284 2 GSLU_RESULT,
0285 2 WORK_AREA : VECTOR [CH$ALLOCATION(50)],
0286 2 WORK_PTR,
0287 2 WORK_LENGTH,
0288 2 HOLD_FLAG,
0289 2 SUBST_INDEX;
0290 2
0291 2 HOLD_FLAG=.KHAR;                                !Remember current <SUBSTITUTE flag>
0292 2
0293 2 KCNS();                                         !Get next character.
0294 2
0295 2 !Make sure user has specified doubled <SUBSTITUTE flag>s.
0296 2 !(Single <SUBSTITUTE flag>s are for extensions, i.e., macro facility).
0297 2 IF .KHAR NEQ .HOLD_FLAG THEN                  !Check for error case first.
0298 2 BEGIN                                            !Not doubled <SUBSTITUTE flag>s
0299 3 ERMS ( RNFFNA
0300 3     ..FS_START (IRA)
0301 3     ,CH$DIFF (.FS_NEXT (IRA), .FS_START (IRA)) );
0302 3
0303 2 RETURN;
0304 2 END;
0305 2
0306 2 !Position to character immediately after the second <SUBSTITUTE flag>
0307 2 KCNS();
0308 2
0309 2 !Fudge up DUMMY FS to look like a FIXED_STRING;
0310 2 FS_MAXSIZE (DUMMY_FS) = 50;
0311 2 FS_INIT (DUMMY_FS);
0312 2
0313 2 !Attempt to pick up the word DATE or TIME.
```

```
; 186      0313 2     GSLU_RESULT = GSLU (IRA, DUMMY_FS);
187      0314 2
188      0315 2     !Make sure something was found.
189      0316 2     IF .GSLU_RESULT NEQ GSLU_NORMAL THEN      ! If nothing was found,
190      0317 2       BEGIN                                ! something's wrong.
191      0318 3       ERMS ( RNFFNA
192      0319 3           .FS_START (IRA)
193      0320 3           .CH$DIFF (.FS_NEXT (IRA), .FS_START (IRA)));
194      0321 3       RETURN;
195      0322 2       END;
196      0323 2
197      0324 2     ! Attempt to identify the word. The result of this block is either a RETURN
198      0325 2     ! if the word was not identified, or an indicator as to which routine to
199      0326 2     ! call to translate the string into something displayable.
200      0327 3     BEGIN
201      0328 3       LOCAL
202      0329 3           PTR;
203      0330 3
204      0331 3     !Assume the word won't be identified.
205      0332 3     SUBST_INDEX = 0;
206      0333 3
207      0334 3     INCR I FROM 0 TO (.SUBST_NAMES[-1]-1) DO
208      0335 4       BEGIN
209      0336 4           IF CHSEQL ( .FS_LENGTH(DUMMY_FS)
210      0337 4               ..FS_START(DUMMY_FS)
211      0338 4               .CH$RCHAR(.SUBST_NAMES[I])
212      0339 4               .CH$PLUS(.SUBST_NAMES[I],1) ) THEN
213      0340 5           BEGIN
214      0341 5               SUBST_INDEX = .I + 1;          !Keyword identified!!!!
215      0342 5               EXITLOOP                      !Remember which word it was.
216      0343 5           END
217      0344 3
218      0345 3
219      0346 3     !See if the keyword was found.
220      0347 3     IF .SUBST_INDEX EQL 0 THEN
221      0348 4           BEGIN
222      0349 4               !Word was not identified
223      0350 4               ERMS (RNFDUS, .FS_START (IRA), .CH$DIFF (.FS_NEXT (IRA), .FS_START (IRA)));
224      0351 3               RETURN
225      0352 2           END;                               !End of identification of word.
226      0353 2
227      0354 2     ! Now get the character string that the user has requested. The result of
228      0355 2     ! this CASE statement is that WORK_AREA will contain the text, text, and
229      0356 2     ! WORK_LENGTH will be the number of characters.
230      0357 2
231      0358 2     WORK_PTR = CH$PTR(WORK_AREA);
232      0359 2     CASE .SUBST_INDEX FROM FIRST_INDEX TO LAST_INDEX OF
233      0360 2       SET
234      0361 2           [SUBST_DATE]:
235      0362 3           BEGIN
236      0363 3               CNVDAT(DATTIM, WORK_PTR, WORK_LENGTH);
237      0364 3               WORK_PTR = CH$PTR(WORK_AREA)
238      0365 2           END;
239      0366 2
240      0367 2           [SUBST_TIME]:
241      0368 3           BEGIN
242      0369 3               CNVTIM(DATTIM, WORK_PTR, WORK_LENGTH);
```

SUBST  
V04-000

Process <SUBSTITUTE flag>  
SUBST --

K 13  
16-Sep-1984 01:52:20  
14-Sep-1984 13:08:17  
VAX-11 Bliss-32 v4.0-742  
DISK\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1 Page 7 (4)

TITI  
V04

```
243 0370 3 WORK_PTR = CH$PTR(WORK_AREA)
244 0371 2 END;
245 0372 2
246 0373 2 [SUBST_YEAR]:
247 0374 3 BEGIN
248 0375 3 WORK_LENGTH = PACBAS (.DATTIM [0], WORK_PTR, 10);
249 0376 3 WORK_PTR = CH$PTR(WORK_AREA);
250 0377 2 END;
251 0378 2
252 0379 2 [SUBST_MONTH]:
253 0380 3 BEGIN
254 0381 3 BIND
255 0382 3 MONTHS = UPLIT (
256 U 0383 3 XIF german %THEN
257 U 0384
258 U 0385
259 U 0386
260 U 0387
261 U 0388
262 U 0389
263 U 0390
264 U 0391
265 U 0392
266 U 0393
267 U 0394
268 U 0395
269 U 0396
270 U 0397 XELSE
271 U 0398 XIF french %THEN
272 U 0399
273 U 0400
274 U 0401
275 U 0402
276 U 0403
277 U 0404
278 U 0405
279 U 0406
280 U 0407
281 U 0408
282 U 0409
283 U 0410 XELSE
284 U 0411 XIF italian %THEN
285 U 0412
286 U 0413
287 U 0414
288 U 0415
289 U 0416
290 U 0417
291 U 0418
292 U 0419
293 U 0420
294 U 0421
295 U 0422
296 U 0423
297 U 0424
298 U 0425
299 U 0426

        WORK_PTR = CH$PTR(WORK_AREA)
        END;

        [SUBST_YEAR]:
        BEGIN
        WORK_LENGTH = PACBAS (.DATTIM [0], WORK_PTR, 10);
        WORK_PTR = CH$PTR(WORK_AREA);
        END;

        [SUBST_MONTH]:
        BEGIN
        BIND
        MONTHS = UPLIT (
XIF german %THEN
CH$PTR(UPLIT(%STRING('Jaenner'))),
CH$PTR(UPLIT(%STRING('Februar'))),
CH$PTR(UPLIT(%STRING('Maerz'))),
CH$PTR(UPLIT(%STRING('April'))),
CH$PTR(UPLIT(%STRING('Mai'))),
CH$PTR(UPLIT(%STRING('Juni'))),
CH$PTR(UPLIT(%STRING('Juli'))),
CH$PTR(UPLIT(%STRING('August'))),
CH$PTR(UPLIT(%STRING('September'))),
CH$PTR(UPLIT(%STRING('Oktober'))),
CH$PTR(UPLIT(%STRING('November'))),
CH$PTR(UPLIT(%STRING('Dezember')))

XELSE
XIF french %THEN
CH$PTR(UPLIT(%STRING('January'))),
CH$PTR(UPLIT(%STRING('February'))),
CH$PTR(UPLIT(%STRING('March'))),
CH$PTR(UPLIT(%STRING('April'))),
CH$PTR(UPLIT(%STRING('May'))),
CH$PTR(UPLIT(%STRING('June'))),
CH$PTR(UPLIT(%STRING('July'))),
CH$PTR(UPLIT(%STRING('August'))),
CH$PTR(UPLIT(%STRING('September'))),
CH$PTR(UPLIT(%STRING('October'))),
CH$PTR(UPLIT(%STRING('November'))),
CH$PTR(UPLIT(%STRING('December')))

XELSE
XIF italian %THEN
CH$PTR(UPLIT(%STRING('January'))),
CH$PTR(UPLIT(%STRING('February'))),
CH$PTR(UPLIT(%STRING('March'))),
CH$PTR(UPLIT(%STRING('April'))),
CH$PTR(UPLIT(%STRING('May'))),
CH$PTR(UPLIT(%STRING('June'))),
CH$PTR(UPLIT(%STRING('July'))),
CH$PTR(UPLIT(%STRING('August'))),
CH$PTR(UPLIT(%STRING('September'))),
CH$PTR(UPLIT(%STRING('October'))),
CH$PTR(UPLIT(%STRING('November'))),
CH$PTR(UPLIT(%STRING('December')))

XELSE
CH$PTR(UPLIT(%STRING('January'))),
CH$PTR(UPLIT(%STRING('February'))),
```

```
300 0427 3 CHSPTR(UPLIT(%STRING('March'))),  
301 0428 3 CHSPTR(UPLIT(%STRING('April'))),  
302 0429 3 CHSPTR(UPLIT(%STRING('May'))),  
303 0430 3 CHSPTR(UPLIT(%STRING('June'))),  
304 0431 3 CHSPTR(UPLIT(%STRING('July'))),  
305 0432 3 CHSPTR(UPLIT(%STRING('August'))),  
306 0433 3 CHSPTR(UPLIT(%STRING('September'))),  
307 0434 3 CHSPTR(UPLIT(%STRING('October'))),  
308 0435 3 CHSPTR(UPLIT(%STRING('November'))),  
309 0436 3 CHSPTR(UPLIT(%STRING('December')))  
310 0437 3  
311 0438 3  
312 0439 3  
313 0440 3 OWN  
314 U 0441 3 XIF german %THEN MONTHL : VECTOR [12] INITIAL (  
315 U 0442 3 7, 7, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8  
316 U 0443 3 XELSE  
317 U 0444 3 XIF french %THEN  
318 U 0445 3 7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8  
319 U 0446 3 XELSE  
320 U 0447 3 XIF italian %THEN  
321 U 0448 3 7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8  
322 U 0449 3 XELSE  
323 U 0450 3 7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8  
324 U 0451 3  
325 U 0452 3  
326 U 0453 3 WORK_PTR = .MONTHS [.DATTIM[1] - 1];  
327 U 0454 3 WORK_LENGTH = .MONTHL [.DATTIM[1] - 1];  
328 U 0455 2 END;  
329 U 0456 2  
330 U 0457 2 [SUBST_DAY]:  
331 U 0458 3 BEGIN  
332 U 0459 3 WORK_LENGTH = PACBAS (.DATTIM [2], WORK_PTR, 10);  
333 U 0460 3 WORK_PTR = CHSPTR(WORK_AREA);  
334 U 0461 2 END;  
335 U 0462 2  
336 U 0463 2 [SUBST_HOURS]:  
337 U 0464 3 BEGIN  
338 U 0465 3 !Assume result will be less than two digits, and prepad with a '0'.  
339 U 0466 3 CHSWCHAR(%C'0', CHSPTR(WORK AREA));  
340 U 0467 3 WORK_PTR = CH$PLUS(.WORK_PTR, 1);  
341 U 0468 3 WORK_LENGTH = PACBAS (.DATTIM [3], WORK_PTR, 10);  
342 U 0469 3 WORK_PTR = CHSPTR(WORK_AREA);  
343 U 0470 3 !If the result was at least two digits, forget the leading zero.  
344 U 0471 3 IF .WORK_LENGTH EQL 1 THEN  
345 U 0472 3 WORK_LENGTH = .WORK_LENGTH + 1  
346 U 0473 3 ELSE  
347 U 0474 3 WORK_PTR = CH$PLUS(.WORK_PTR, 1);  
348 U 0475 2 END;  
349 U 0476 2  
350 U 0477 2 [SUBST_MINUTES]:  
351 U 0478 3 BEGIN  
352 U 0479 3 !Assume result will be less than two digits, and prepad with a '0'.  
353 U 0480 3 CHSWCHAR(%C'0', CHSPTR(WORK AREA));  
354 U 0481 3 WORK_PTR = CH$PLUS(.WORK_PTR, 1);  
355 U 0482 3 WORK_LENGTH = PACBAS (.DATTIM [4], WORK_PTR, 10);  
356 U 0483 3 WORK_PTR = CHSPTR(WORK_AREA);
```

SUBST  
V04-000

Process <SUBSTITUTE flag>  
SUBST --

M 13  
16-Sep-1984 01:52:20 VAX-11 Bliss-32 v4.0-742  
14-Sep-1984 13:08:17 DISK\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1 Page 9  
(4)

357 0484 3 !If the result was at least two digits, forget the leading zero.  
358 0485 3 IF .WORK\_LENGTH EQL 1 THEN  
359 0486 3 WORK\_LENGTH = .WORK\_LENGTH + 1  
360 0487 3 ELSE  
361 0488 3 WORK\_PTR = CH\$PLUS(.WORK\_PTR, 1);  
362 0489 2 END;  
363 0490 2  
364 0491 2 [SUBST\_SECONDS]:  
365 0492 3 BEGIN  
366 0493 3 !Assume result will be less than two digits, and prepad with a '0'.  
367 0494 3 CH\$WCHAR(%C'0', CH\$PTR(WORK\_AREA));  
368 0495 3 WORK\_PTR = CH\$PLUS(.WORK\_PTR, 1);  
369 0496 3 WORK\_LENGTH = PACBAS (.DATTIM [5], WORK\_PTR, 10);  
370 0497 3 WORK\_PTR = CH\$PTR(WORK\_AREA);  
371 0498 3 !If the result was at least two digits, forget the leading zero.  
372 0499 3 IF .WORK\_LENGTH EQL 1 THEN  
373 0500 3 WORK\_LENGTH = .WORK\_LENGTH + 1  
374 0501 3 ELSE  
375 0502 3 WORK\_PTR = CH\$PLUS(.WORK\_PTR, 1);  
376 0503 2 END;  
377 0504 2 TES:  
378 0505 2  
379 0506 2 !Now feed these characters into the document.  
380 0507 2 INCR I FROM 1 TO .WORK\_LENGTH DO  
381 0508 2 ENDCHR ( CH\$RCHAR\_A (WORK\_PTR) );  
382 0509 2  
383 0510 2 !That's all there is, so just return.  
384 0511 2 RETURN;  
385 0512 1 END;

!End of SUBST

.TITLE SUBST Process <SUBSTITUTE flag>  
.IDENT \V04-000\

.PSECT SPLITS,NOWRT,NOEXE,2

00 00 00 45 54 41 44 04 00000 P.AAB:	.ASCII <4>\DATE\<0><0><0>
00 00 00 45 4D 49 54 04 00008 P.AAC:	.ASCII <4>\TIME\<0><0><0>
00 00 00 52 41 45 59 04 00010 P.AAD:	.ASCII <4>\YEAR\<0><0><0>
00 00 48 54 4E 4F 4D 05 00018 P.AAE:	.ASCII <5>\MONTH\<0><0>
00 00 53 52 55 4F 48 05 00020 P.AAF:	.ASCII <3>\DAY\
53 45 54 55 4E 49 4D 07 00024 P.AAG:	.ASCII <5>\HOURS\<0><0>
53 44 4E 4F 43 45 53 07 0002C P.AAH:	.ASCII <7>\MINUTES\
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 000040 P.AAA:	.ASCII <7>\SECONDS\
00000000' 00000000' 00000000' 00000000' 00000000' 000058	.LONG 8
00 79 72 61 75 6E 61 4A 00060 P.AAK:	.ADDRESS P.AAB, P.AAC, P.AAD, P.AAE, P.AAF, - P.AAG, P.AAH, P.AAI
79 72 61 75 72 62 65 46 00068 P.AAL:	.ASCII \January\<0>
00 00 00 68 63 72 61 4D 00070 P.AAM:	.ASCII \February\
00 00 00 6C 69 72 70 41 00078 P.AAN:	.ASCII \March\<0><0><0>
00 00 00 6C 69 72 70 41 00080 P.AAO:	.ASCII \April\<0><0><0>
00 00 00 6C 69 72 70 41 00084 P.AAP:	.ASCII \May\<0>
00 00 00 6C 69 72 70 41 00088 P.AAQ:	.ASCII \June\
00 00 00 72 00 00 74 73 75 67 75 41 0008C P.AAR:	.ASCII \July\
00 00 00 72 65 62 6D 65 74 70 65 53 00094 P.AAS:	.ASCII \August\<0><0>
00 00 00 72 65 62 6F 74 63 4F 000A0 P.AAT:	.ASCII \September\<0><0><0>
00 00 00 72 65 62 6F 74 63 4F 000A0 P.AAT:	.ASCII \October\<0>

SUBST  
V04-000Process <SUBSTITUTE flag>  
SUBST --N 13  
16-Sep-1984 01:52:20 VAX-11 Bliss-32 v4.0-742  
14-Sep-1984 13:08:17 DISK\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1 Page 10 (4)TITI  
V04-

72 65 62 6D 65 76 6F 4E 000A8 P.AAU: .ASCII \November\  
 72 65 62 6D 65 63 65 44 000B0 P.AAV: .ASCII \December\  
 00000000: 00000000: 00000000: 00000000: 00000000: 000B8 P.AAJ: .ADDRESS P.AAK, P.AAL, P.AAM, P.AAN, P.AAO, -  
 00000000: 00000000: 00000000: 00000000: 00000000: 000D0 P.AAP, P.AAQ, P.AAR, P.AAS, P.AAT, P.AAU, -  
 P.AAV

.PSECT \$OWNS,NOEXE,2

00000004 00000003 00000005 00000005 00000008 00000007 00000004 00000000 MONTHL: .LONG 7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8

00000008 00000008 00000007 00000009 00000006 00000004 000018

SUBST NAMES= P.AAA  
 MONTHS= P.AAJ  
 .EXTRN DATTIM, IRA, KHAR  
 .EXTRN RNFFNA, RNFUDS, CNVDAT  
 .EXTRN CNVTIM, ENDCHR, ERMS  
 .EXTRN GSLU, PACBAS, RINTES

.PSECT \$CODE\$,NOWRT,2

5A 00000000G		EF 07FC 00000	.ENTRY SUBST, Save R2,R3,R4,R5,R6,R7,R8,R9,R10	: 0257
59 00000000		EF 9E 00002	MOVAB PACBAS, R10	
58 00000000G		EF 9E 00009	MOVAB SUBST_NAMES, R9	
57 00000000G		EF 9E 00010	MOVAB KHAR, R8	
56 00000000G		EF 9E 00017	MOVAB DATTIM, R7	
5E 80		AE 9E 0001E	MOVAB IRA+12, R6	
50		68 D0 00025	MOVAB -128(SP), SP	
		66 D5 0002C	MOVL KHAR, HOLD_FLAG	
		09 14 0002E	TSTL IRA+12	
68	00G	8F 9A 00030	BGTR 1\$	
66		01 CE 00034	MOVZBL #RINTES, KHAR	
		09 11 00037	MNEGL #1, IRA+12	
68	F8	B6 9A 00039	BRB 2\$	
	F8	A6 D6 0003D	1\$: MOVZBL @IRA+4, KHAR	
		66 D7 00040	INCL IRA+4	
50		68 D1 00042	2\$: DECL IRA+12	
		36 12 00045	CMPL KHAR, HOLD_FLAG	
		66 D5 00047	BNEQ 5\$	
		09 14 00049	TSTL IRA+12	
68	00G	8F 9A 0004B	BGTR 3\$	
66		01 CE 0004F	MOVZBL #RINTES, KHAR	
		09 11 00052	MNEGL #1, IRA+12	
68	F8	B6 9A 00054	BRB 4\$	
	F8	A6 D6 00058	3\$: MOVZBL @IRA+4, KHAR	
		66 D7 0005B	INCL IRA+4	
44	AE	32 7D 0005D	4\$: DECL IRA+12	
3C	AE	4C AE 9E 00061	MOVQ #50, DUMMY FS+8	
40	AE	3C AE D0 00066	MOVAB DUMMY_FS+16, DUMMY_FS	
		3C AE 9F 0006B	MOVL DUMMY_FS, DUMMY_FS+4	
		F4 A6 9F 0006E	PUSHAB DUMMY_FS	
00000000G	EF	02 FB 00071	PUSHAB IRA	
01		50 D1 00078	CALLS #2, GSLU	
		11 13 0007B	CMPL GSLU_RESULT, #1	
7E	F8	A6 C3 0007D	5\$: BEQL 6\$	
		F4 A6 DD 00083	SUBL3 IRA, IRA+4, -(SP)	
		00000000G 8F DD 00086	PUSHL IRA	
			PUSHL #RNFFNA	

SUBST  
V04-000

Process <SUBSTITUTE flag>  
SUBST --

8 14  
16-Sep-1984 01:52:20 VAX-11 Bliss-32 V4.0-742 Page 11  
14-Sep-1984 13:08:17 DISK\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1 (4)

SUBST  
V04-000Process <SUBSTITUTE flag>  
SUBST --C 14  
16-Sep-1984 01:52:20  
14-Sep-1984 13:08:17 VAX-11 Bliss-32 v4.0-742  
DISK\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1

Page 12 (4)

TITI  
V04-

		08	AE	9F	00141	PUSHAB	WORK_PTR				
		0C	A7	DD	00144	PUSHL	DATTIM+12				
			0F	11	00147	BRB	22\$				
	08	AE	30	90	00149	21\$:	MOVB	#48, WORK_AREA			
		04	AE	D6	0014D	INCL	WORK_PTR	0480			
			0A	DD	00150	PUSHL	#10	0481			
		08	AE	9F	00152	PUSHAB	WORK_PTR	0482			
		10	A7	DD	00155	PUSHL	DATTIM+16				
			03	FB	00158	22\$:	CALLS	#3, PACBAS			
	04	AE	50	D0	0015B	MOVL	R0, WORK_LENGTH	0483			
		01	AE	9E	0015E	MOVAB	WORK_AREA, WORK_PTR	0485			
			6E	D1	00163	CMPL	WORK_LENGTH, #1				
			21	13	00166	BEQL	24\$				
	08	AE	23	11	00168	BRB	25\$	0488			
			30	90	0016A	23\$:	MOVB	#48, WORK_AREA			
		04	AE	D6	0016E	INCL	WORK_PTR	0494			
			0A	DD	00171	PUSHL	#10	0495			
		08	AE	9F	00173	PUSHAB	WORK_PTR	0496			
		14	A7	DD	00176	PUSHL	DATTIM+20				
			03	FB	00179	CALLS	#3, PACBAS				
	04	AE	50	D0	0017C	MOVL	R0, WORK_LENGTH	0497			
		01	AE	9E	0017F	MOVAB	WORK_AREA, WORK_PTR	0499			
			6E	D1	00184	CMPL	WORK_LENGTH, #1				
			04	12	00187	BNEQ	25\$				
			6E	D6	00189	24\$:	INCL	WORK_LENGTH			
			03	11	0018B	BRB	26\$	0500			
		04	AE	D6	0018D	25\$:	INCL	WORK_PTR			
			52	D4	00190	26\$:	CLRL	I			
			0E	11	00192	BRB	28\$	0502			
		EE	0000000G	7E	04	BE	9A	00194	27\$:	MOVZBL	@WORK_PTR, -(SP)
					08	AE	D6	00198	INCL	WORK_PTR	0507
					01	FB	0019B	CALLS	#1, ENDCHR	0508	
					6E	F3	001A2	AOBLEQ	WORK_LENGTH, I, 27\$		
					04	001A6	RET			0512	

: Routine Size: 423 bytes, Routine Base: \$CODE\$ + 0000

: 386 0513 1  
: 387 0514 1 END  
: 388 0515 0 ELUDOM

!End of module

## PSECT SUMMARY

Name	Bytes	Attributes
\$SPLITS	232	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$DOWNS	48	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODES	423	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

SUBST  
V04-000

Process <SUBSTITUTE flag>  
SUBST --

D 14  
16-Sep-1984 01:52:20  
14-Sep-1984 13:08:17

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1

Page 13  
(4)

TIT  
V04

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	0	0	252	00:00.2
-\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	13	1	86	00:00.2

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:SUBST/OBJ=OBJ\$:SUBST MSRC\$:SUBST/UPDATE=(ENH\$:SUBST)

: Size: 423 code + 280 data bytes

: Run Time: 00:10.8

: Elapsed Time: 00.33.1

: Lines/CPU Min: 2850

: Lexemes/CPU-Min: 21154

: Memory Used: 131 pages

: Compilation Complete

0349 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

